
Orientação a Objetos

Programação em C++

Referências

Grupo de Slides 1 – Parte C

Prof. Dr. Jean Marcelo SIMÃO – DAINF / UTFPR

Monitor: Vitor C. M. Corrêa – discente de Engenharia de Computação DAINF / DAELN

Referências

Podemos dividir os tipos de variáveis simples da linguagem C++ em três grandes grupos:

Variáveis normais, que guardam valores diretamente.

Ponteiros, que guardam os endereços de outra variável e podem ser utilizados para obter o valor de onde apontam.

E as *Referências*, que serão abordadas nesse grupo de slides.

Referências

Podemos imaginar a referência como um tipo de variável que age como um pseudônimo, uma espécie de “segundo” nome, para outra variável ou valor.

Criação de uma referência:

```
int valor = 5;  
int &ref = valor;
```

Referências

Mas como a referência funciona?

Ela irá agir identicamente à variável que está referenciando, como se fosse a própria variável.

```
int valor = 5;           // int normal
int &ref = valor;       // referência à variável valor
valor = 6;              // valor agora é igual a 6
ref = 7;                // valor agora é igual a 7
std::cout << valor;    // imprime 7
++ref;                  // incrementa a variável valor
std::cout << valor;    // imprime 8
std::cout << &valor;   // imprime 0xAABBCCDD (por exemplo)
std::cout << &ref;     // também imprime 0xAABBCCDD
```

Referências

Mas como a referência funciona?

Diferentemente de um ponteiro, uma referência deve sempre conter um valor que é atribuído em sua inicialização, e que não pode ser mudado.

```
int valor1 = 5;
int &ref1 = valor1; // ref1 agora referência variável valor1
int &ref2 = 6;      // causa um erro
int valor2 = 7;
int &ref3 = valor1; // ref3 agora é uma referência para valor1
ref3 = valor2      // atribui valor2 à variável referenciada por ref3 (valor1)
```

Referências

E quanto a sua utilização na prática?

Na prática, em C++, referências são utilizadas quase que exclusivamente para facilitar a utilização de funções, sejam como parâmetros ou como retorno.

Quando utilizada como tipo de parâmetro de uma função, temos o que é conhecido como referência escondida. Por sua vez, quando utilizada como tipo de retorno, daí é possível fazer um encadeamento de chamadas.

Referência escondida

O termo referência escondida se refere à quando o tipo de um ou mais parâmetros de uma função é do tipo *referência*. Assim, o parâmetro irá referenciar a variável passada à função e poderá modificá-la.

Exemplo:

```
void alteraNumero ( int& num, int novoNum )
{
    num = novoNum; // a variável "num" é passado por referência, já "novoNum" por valor
}
```

```
int numero = 5;           // o inteiro 5 é atribuído à variável numero
alteraNumero( numero, 3 ); // a variável "numero" agora é modificada e vale 3
std::cout << numero;     // imprime 3
alteraNumero( numero, 7 ); // a variável "numero" agora é modificada e vale 7
std::cout << numero;     // imprime 7
```

Referência escondida

A utilização da referência escondida é extremamente útil e bastante utilizado na prática, porém deve-se atentar à chamada da função, visto que a função pode modificar a variável passada de parâmetro de maneira indesejada.

Devido à isso, pertinente salientar a importância da documentação de qualquer código escrito.

Quanto ao encadeamento, o assunto será visto no tópico de sobrecarga de operadores.